ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/257435951

An efficient hybrid learning algorithm for neural network–based speech recognition systems on FPGA chip

Article *in* Neural Computing and Applications · June 2013

DOI: 10.1007/s00521-013-1428-5

CITATIONS
7

READS
106

2 authors, including:

Shing-Tai Pan
National University of Kaohsiung
93 PUBLICATIONS **632** CITATIONS

SEE PROFILE

All content following this page was uploaded by Shing-Tai Pan on 23 March 2016.

The user has requested enhancement of the downloaded file.

Neural Comput & Applic
DOI 10.1007/s00521-013-1428-5

ORIGINAL ARTICLE

An efficient hybrid learning algorithm for neural network–based speech recognition systems on FPGA chip

Shing-Tai Pan · Min-Lun Lan

Received: 11 June 2012 / Accepted: 20 May 2013
© Springer-Verlag London 2013

Abstract This paper implemented an artificial neural network (ANN) on a field programmable gate array (FPGA) chip for Mandarin speech measurement and recognition of nonspecific speaker. A three-layer hybrid learning algorithm (HLA), which combines genetic algorithm (GA) and steepest descent method, was proposed to fulfill a faster global search of optimal weights in ANN. Some other popular evolutionary algorithms, such as differential evolution, particle swarm optimization and improve GA, were compared to the proposed HLA. It can be seen that the proposed HLA algorithm outperforms the other algorithms. Finally, the designed system was implemented on an FPGA chip with an SOC architecture to measure and recognize the speech signals.

Keywords Field programmable gate array · Genetic algorithm · Hybrid learning algorithm · Speech recognition

1 Introduction

The measurement and recognition of speech signal are explored in many researches [1–3]. The most popular methods for speech recognition are artificial neural network (ANN) [4] and hidden Markov model (HMM) [5]. Recently, the support vector machine (SVM) incorporated with HMM is applied on this field and has obtained some impressive results. Furthermore, due to the marvelous improvement on the manufacture of ICs in recent years, the capacity and the calculation ability of a chip are dramatically increased. Consequently, the speech measurement and recognition systems can be implemented in a chip rather than in a computer. This makes the embedded speech recognition systems in many consuming electronics realizable. In paper [6], a micro-controller 8051 chip is adopted for implementation of a speech recognition system. However, in order to have more extensive application of speech recognition in consuming electronics, a platform with better computational ability and flexibility, such as field programmable gate array (FPGA)-based embedded systems with SOC structure, is adopted in this paper for the implementation of the systems. On the embedded platform, the speech recognition systems can perform much more applications by combing the other units on the platform. This is the reason why the FPGA-based embedded platform is adopted in this paper to realize the speech recognition systems. Due to the above reason and the fact that ANN surpasses the other methods in computing speed, ANN is more suitable for chip implementation. Therefore, this study adopts ANN for speech recognition instead of HMM and SVM.

However, since there are enormous parameters including the weights and biases in ANN, it is difficult to train ANN well. Consequently, the traditional training method, steepest descent method (SDM), cannot efficiently train ANN due to the high degree of nonlinearity of the systems. Recently, with the enormously enhanced computation ability of PC, the evolutionary algorithms have widely been applied on many applications. The most popular evolutionary algorithm is genetic algorithm (GA) [7, 8]. However, although GA converges in many applications to a good result, the convergence rate is always slow. Hence, it

S.-T. Pan (✉)
Department of Computer Science and Information Engineering, National University of Kaohsiung, No. 700, Kaohsiung University Rd., Nanzih Dist., Kaohsiung 811, Taiwan, ROC
e-mail: stpan@nuk.edu.tw

M.-L. Lan
Institute of Computer Science and Information Engineering, Shu-Te University, Kaohsiung 824, Taiwan, ROC

Published online: 15 June 2013

is important to find an algorithm with fast convergence rate to train ANN. In this paper, a more efficient algorithm by combining GA and SDM is proposed to train ANN. Besides, since the evolutionary algorithms, improve GA (IGA), particle swarm optimization (PSO) and differential evolution (DE), are widely applied in many applications, this paper will compare the performance between these algorithms and the proposed algorithm. The experimental results will show that the proposed hybrid learning algorithm (HLA) outperforms other algorithms.

The organization of this paper is as follows. In Sect. 2, the pre-processes for speech sound signals are introduced. Section 3 briefly describes the architecture of ANN-based speech recognition systems. Some popular evolutionary algorithms and the proposed HLA algorithm are introduced in Sect. 4. In this section, the ways of training ANN for speech recognition are explored. Section 5 shows the experimental results of the proposed algorithm and compares these results to various algorithms to show the advantage of the proposed algorithm. Moreover, the implementation of the proposed speech recognition system on an FPGA-based embedded system with SOC architecture is explored. Some experiments of speech recognition on the hardware are then revealed. Finally, some conclusions are given in Sect. 6.

2 Pre-processing for speech sound signals

Prior to the speech recognition for some speech signals, we will do some pre-processing on the speech sound signals, such as endpoint detection, pre-emphasis, multiplication of Hamming window and retrieval of features. Since the original speech sound signal itself belongs to time-varying signal and is complicated, the signals need to be sliced into many small frames. When a speech sound signal is cut into small frames, the frames (audio frame) can be viewed as time-invariant signal [9]. After the speech sound is divided into several audio frames, the feature is then retrieved by using MFCC for each frame and then is used for train and recognition. Some steps of pre-processing for the speech sounds are described as follows for the purpose of clarification.

2.1 Pre-emphasis

All the speech sound signals are fed through into a high-pass filter to recover the reduced signal. The difference equation governing the high-pass filter is as follows:

$$S(n) = X(n) - 0.95X(n-1), \quad 1 \le n \le L; \tag{1}$$

In the Eq. (1), $S(n)$ represents the signal that has been processed with pre-emphasis, while $X(n)$ represents the original signal, and L is the length (number of sampling) of each audio frame.

2.2 Taking Hamming window

The following Eqs. (2) and (3) describe the Hamming window used in this paper for speech signals [10]:

$$W(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2n\pi}{L-1}\right), & 0 \le n \le L-1; \\ 0, \text{otherwise}; \end{cases} \tag{2}$$

$$F(n) = W(n) \times S(n); \tag{3}$$

in which L is the length of audio frame; $S(n)$ is a frame of speech signal; $W(n)$ is the Hamming window and $F(n)$ is the result of speech signal multiplied by Hamming window.

2.3 Retrieval of feature

For speech recognition, the methods commonly used for extracting the feature of sound can be divided into two categories: one is time-domain method and the other is frequency-domain method. The way of time-domain method is more direct and time saving with fewer operations. On the other hand, in frequency-domain method, it is necessary to take Fourier transform on the speech signals. This process causes more operations and makes the computation of feature becomes more complicated. Consequently, it leads to the requirement of much more computation time compared to time-domain method. In time domain, the most popular method for features extraction is linear predict coding, while in frequency domain, the most popular methods are Cepstrum coefficient and MFCC [10]. Because MFCC is more close to the distinction made by human ears on speech sound, it is used in this paper to extract the features for speech sound signals. The processes of MFCC are described as follows. First, each audio frame is transformed to frequency domain, says $|X(k)|$. The energy $Y(m)$ is then obtained from multiplying each frequency domain $|X(k)|$ by a triangle filter as follows:

$$B_m(k) = \begin{cases} 0, & k < f_{m-1} \\ \frac{k - f_{m-1}}{f_m - f_{m-1}}, & f_{m-1} \le k \le f_m \\ \frac{f_{m+1} - k}{f_{m+1} - f_m}, & f_m \le k \le f_{m+1} \\ 0, & f_{m+1} < k \end{cases} \tag{4}$$

where $1 \le m \le M$ and M is the number of the filters. After accumulating and applying the $log(.)$ function, the energy function is got as

$$Y(m) = \log\left\{ \sum_{k=f_{m-1}}^{f_{m+1}} |X(k)| B_m(k) \right\}. \tag{5}$$

Applying the discrete cosine transform to M pieces of $Y(m)$, we then obtain

$$c_x(n) = \frac{1}{M}\sum_{m=1}^{M} Y(m)\cos\left(\frac{\pi n\left(m-\frac{1}{2}\right)}{M}\right)$$

in which $c_x(n)$ is the MFCC. In order to decrease the input number of ANN, the first 10 coefficients are used for speech recognition in this paper.

3 ANN-based speech recognition

After retrieving the features of speech sound, the speech signal is then recognized by the following processes. As that we have introduced in the Introduction of this paper, there are many ways for speech recognition through the speech features. For the method of DTW, it has to compare the data in the database of speech sound samples with the tested speech sounds one by one. Consequently, more tested speech sounds will cost more computing time for recognition. Similarly, the process of the method HMM needs much statistical computation for speech recognition. It is seen that the more speeches to be recognized, the more statistical computation must be done. In contrast, for the method of ANN, as long as the training of ANN is completed, the time for the speech recognition will not increase significantly. This is because that the dimension of ANN is fixed after it had been trained and is independent to the amount of the speech to be recognized. Consequently, the advantage of the method ANN for speech recognition is that it can get a faster recognition speed. Hence, this paper adopts the method of ANN for the purpose of chip realization of speech recognition system. In addition to the advantage of low computation time, ANN has another advantage on fault-tolerant capacity which is a awesome property of ANN.

The principles of ANN are all based on multiple-layer perceptron as its systematic structure, and the proposed HLA algorithm is used to train ANN. The multiple-layer structure in the model of multiple-layer perceptron means that it comprises multiple-layered neurons, while the way to transmit signals between every two neurons is just like the situation in single layer. In this paper, we adopt the three-layered structure (i.e., input layer, hidden layer and output layer) of ANN for speech recognition. The structure of ANN is shown in Fig. 1.

In Fig. 1, P_n means the nth input, while $w_{i,j}^k$ represents the weight value of the k layer, and i represents the number of the neuron in $(k-1)$th layer, j is the number of neutron in kth layer; o_n^k represents the nth output of the kth layer, while b_n^k is the bias in the kth layer of nth neuron. The most important goal of back-propagation algorithm is to adjust the weight of ANN through the error function between the output and the target. And then, the modified value will be

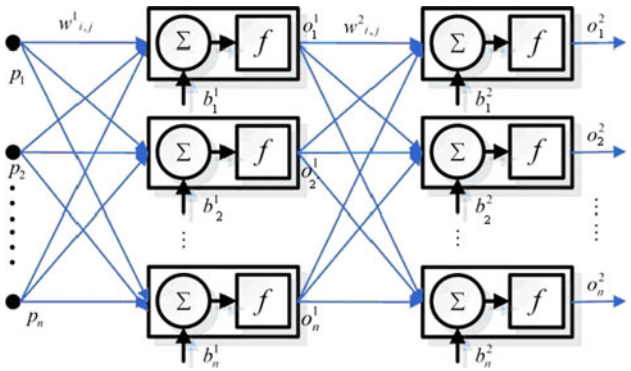

Fig. 1 The structure of artificial neural network (ANN)

transmitted to the neuron in the front layer. This process will continue until the ideal target is achieved. The Eqs. (6) and (7) define the output function of the ANN in this research, in which the variable i represents the number of the neuron in the k-1 layer.

$$o_n^k = f\left(\sum_{\forall i} w_{i,n}^k o_i^{k-1} - b_n^k\right) \tag{6}$$

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}}. \tag{7}$$

4 Train ANN with various algorithms

This section introduces the procedure to train ANN with various algorithms including the proposed HLA algorithm.

4.1 Proposed HLA algorithms

In this section, by exploiting the characteristic that the training performance of SDM and GA complement to each other, a three-stage HLA architecture integrating SDM and GA is proposed and depicted in Fig. 2. Each stage of the proposed HLA is illustrated as follows:

1. *Stage 1* The first stage is to search a better set of initial values for the weights and biases in ANN through SDM.
2. *Stage 2* Based on the initial values obtained from the first stage, GA is then used to make a global search of the weights and biases which minimize the error function of the system.

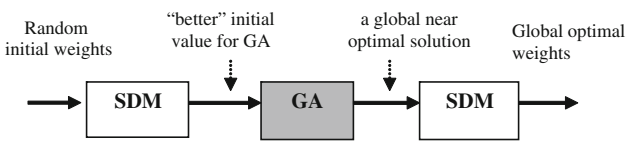

Fig. 2 Three stages of the proposed hybrid learning algorithm (HLA)

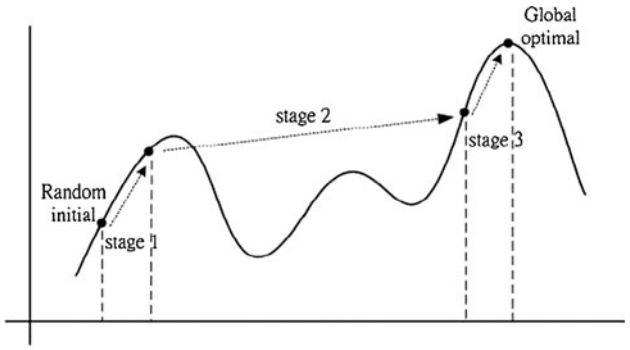

Fig. 3 An illustration of the proposed HLA

3. *Stage 3* In the final stage, in order to speed up the convergence rate of the HLA algorithm, SDM is used again to search the final optimal solution of weights.

It can be seen that SDM is used to speed up the training performance at the head and the tail of the proposed HLA. Figure 3 illustrates the process of the three stages. In the following section, it will be demonstrated that the proposed HLA outperforms other compared algorithms.

4.2 Introduction of the compared algorithms

In order to verify the performance of the proposed HLA, various algorithms IGA [11], PSO [12] and DE [13] are compared and are briefly introduced as follows.

4.2.1 DE

The evolution of DE is described as follows [13]:

$$\underline{v}_G = \underline{x}_{r_1,G} + \lambda \cdot \left(\underline{x}_{best,G} - \underline{x}_{r_1,G}\right) + F \cdot \left(\underline{x}_{r_2,G} - \underline{x}_{r_3,G}\right), \quad (8)$$

where $\underline{v}_G$ and $\underline{x}_{i,G}$, are the vectors of next and current generation, respectively, λ is a control variable, $\underline{x}_{best,G}$ is the best vector in generation G, r_1, r_2, $r_3 \in [0, NP-1]$, $F > 0$ is a real parameter. The value of F mostly falls in the interval $(0, 2]$.

4.2.2 PSO

The evolution of PSO is described as follows [12]:

$$\begin{aligned}\text{vel}[n] = {} & w_p * \text{vel}[n] + c_1 * \text{rand}(n) * (p\text{best}[n] \\ & - \text{particle}[n]) + c_2 * \text{rand}(n) * (g\text{best}[n] \\ & - \text{particle}[n])\end{aligned} \quad (9)$$

$$\text{particle}[n+1] = \text{particle}[n] + \text{vel}[n] \quad (10)$$

where $p$best[] is the best position for a particle, $g$best[] is the best position for all particles in all generations, vel[] is the velocity of a particle, w_p is the inertial weights, c_1 and c_2 are learning factors, particle[] is the position of a particle and rand() is a random number between 0–1.

4.2.3 IGA

The evolution of IGA is described as follows [11]:

$$os_c^1 = [os_1^1 \quad os_2^1 \ldots\ldots \quad os_{no\_vaess}^1] = (P_1 + P_2)/2 \quad (11)$$

$$\begin{aligned} os_c^2 &= [os_1^2 \quad os_2^2 \ldots\ldots \quad os_{no\_\text{vars}}^2] \\ &= P_{\max}(1-w) + \max(P_1, P_2)w \end{aligned} \quad (12)$$

$$\begin{aligned} os_c^3 &= [os_1^3 \quad os_2^3 \ldots \quad os_{no\_\text{vars}}^3] \\ &= P_{\min}(1-w) + \min(P_1, P_2)w \end{aligned} \quad (13)$$

$$\begin{aligned} os_c^4 &= [os_1^4 \quad os_2^4 \ldots \quad os_{no\_\text{vars}}^4] \\ &= \frac{[(P_{\max} + P_{\min})(1-w) + (P_1 + P_2)w]}{2} \end{aligned} \quad (14)$$

where $P_{\max} = [p_{\max}^1 \ \cdots \ p_{\max}^{no\_\text{vars}}]$, $P_{\min} = [p_{\min}^1 \ \cdots \ p_{\min}^{no\_\text{vars}}]$, $os_c^1 \sim os_c^4$ are the chromosomes of the next generation, P_1 and P_2 are the two chromosomes in the parent, $\max(P_1, P_2)$ and $\min(P_1, P_2)$ are the chromosomes in which genes are the maximum and minimum, respectively, of the genes at the corresponding position of the two chromosomes P_1 and P_2. $p_{\max}^i, p_{\min}^i$ are the upper bound and lower bound of ith genes in the search space. The parameter$w \in [0, 1]$ is a real number.

The choice of the parameters for the various algorithms in this experiment is introduced in Table 3 in next section. All the parameters are set according to the suggestions from the papers [11–13].

5 Experimental results

This section shows the experiment results of ANN-based speech recognition. Besides, an FPGA-based SOPC system is used to implement the designed system.

In this experiment, a fixed number of audio frames are adopted to meet the fixed input number of ANN. Therefore, a dynamic overlap rate is used and derived as follows [14]:

$$R = \frac{1}{l_F}\left\{l_F - \text{floor}\left[\frac{l_S - l_F}{N_F - 1}\right]\right\} \quad (15)$$

in which R is the overlap rate of two adjacent frames, l_F is the length of a frame, l_S is the total sampling points, N_F is the number of frames in a speech, floor(x) is the maximum integer that is smaller than x. As for the endpoint detection (EPD) in this experiment, the time-domain EPD is used for reducing the computation time. The threshold value for EPD in Eq. (16) is obtained by calculating the average energy value of the preceding silence frames (background noise), then added with 7.5 % of the maximum energy of all frames [14, 15].

$$\text{Threshold} = 7.5\% \times \max[E(n)] + \frac{1}{K}\sum_{i=1}^{K} E(i); 1 \le n \le N_F \quad (16)$$

in which K is the number of silence frames, $E(i)$ is the energy of the ith silence frame. In this experiment, K was set to 5. Moreover, after the endpoint detection, the processes of pre-emphasis, taking Hamming windows, are then applied. Thereafter, the features of each frame are obtained by using MFCC described in Sect. 2. In this experiment, the features are with 10 coefficients.

5.1 Training ANN

In this experiment, the speeches 0–9, which are commonly used in the speech recognition research for Mandarin language, are recognized. Forty data for each speech are recorded from ten subjects. This means that there are totally four hundred data for this experiment. For the holdout experiment, 75 % of the data are used for training, while the remaining for testing. Each speech signal is divided into 20 frames with variable overlap rate. Each frame has 10 features derived from 256 sampling points. Hence, the ANN in this experiment has 200 inputs in the input layer and 10 outputs in the output layer. The number of neurons in the hidden layer is determined by an experiment with various neuron numbers. The results are shown in Table 1. The neuron number 30 with highest recognition rate is adopted.

The specs for recording speech are 32 bits length for each sample with 8 kHz sampling rate and a mono channel. The features of the speech signal are obtained from MFCC [10] in a specific frequency band. The specific frequency band is determined by an experiment with various frequency sub-band of the frequency band 100 Hz–4 kHz. The results are shown in Table 2. Table 2 reveals that the best frequency band and best feature order are 133–3.8 kHz and 10, respectively. Hence, in this experiment, these best specs are adopted.

To train ANN, the error function is defined as follows:

$$MSE = \frac{1}{10 \times 10 \times 4} \sum_{k=1}^{10} \sum_{i=0}^{9} \sum_{t=1}^{4} \left| E_{k,i,t} \right|^2 \tag{17}$$

Table 1 Experiments for determining the number of neurons in hidden layer

No. of neurons	20	25	30	35	40
Speech recognition rate	88 %	89 %	91 %	91 %	90 %

in which $E_{i,k,t}$ means the output error for tth record of ith literal by kth person (Table 3).

5.2 Numerical results on PC

The experiment results on the convergence of various algorithms are shown in Fig. 4. From the figure, the final MSE for GA, SDM, DE, IGA, PSO and the proposed HLA are 4.9, 6.5, 8.5, 6.8, 4.02 and 0.01, respectively. It is obvious that the proposed algorithm HLA performs much better than other algorithms. Moreover, the time for training ANN and the speech recognition rates of various algorithms are compared in Table 4. According to Table 4, the proposed HLA surpasses other algorithms both in training time and in speech recognition rate. This is due to the fact that the complementation of the characteristics of SDM and GA is exploited in the proposed HLA. We use SDM algorithm in the head and tail of HLA to speed up the training process and use GA in the middle stage for a global optimal search. This also means that the strategy in the proposed HLA performs well, in which only traditional GA and SDM are used. Moreover, it is worthwhile to note that the recognition rate of ANN trained by DE and PSO is worse and cannot be improved even with more training time. In fact, PSO and DE are only suitable to solve the optimal problem with a small number of parameters. They always got a local optimal solution to the problem with a large number of parameters like the case in this study.

Besides, the comparison between different combination of GA and SDM is depicted in Fig. 5. From Fig. 5, it is obvious that the proposed HLA converges faster than another combination. Moreover, the MSE which the proposed HLA converges to is also better than that of the other method. This proves the advantage of the proposed HLA.

5.3 Realization on FPGA-based embedded hardware

In this experiment, we recorded the recognition results by repeating the testing process 100 times for each speech. Five peoples pronounced each speech 20 times and recorded the recognition results from the FPGA board. Moreover, the Nios II O.S. was downloaded to the FPGA chip in the experiment. The bit length of the registers in the experiment is 32 bits, and the record rate is 8 kHz. The trained speech recognition system is implemented on an embedded platform with SOC architecture that is depicted

Table 2 Experiment of speech recognition rate for various frequency sub-band

Order of feature	6	7	8	9	10	11
Frequency range	265–3 K	232–3.2 K	199–3.4 K	166–3.6 K	133–3.8 K	100–4 K
Recognition rate	73 %	83 %	87 %	90 %	95 %	95 %

Table 3 Choice of the parameters for various algorithms

SDM	GA		DE [13]	PSO [12]	IGA [11]
Learning factor: adaptive	Crossover: linear	Mutation rate: adaptive	$\lambda = 0.2$, $F = 0.5$	$c_1 = c_2 = 2$	w: random

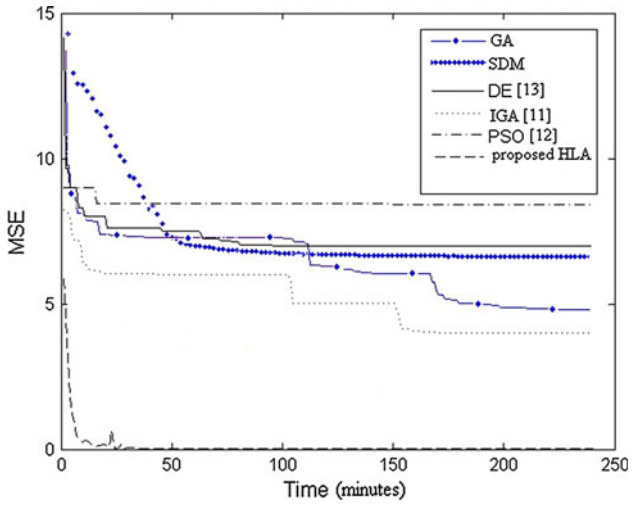

Fig. 4 Convergence of various algorithms

Table 4 Recognition rate and time of ANN trained by various algorithms

Algorithms	PSO [12]	DE [13]	IGA [11]	HLA
Recognition rates	8 %	9 %	95 %	95 %
Time cost for training	48 h	48 h	62 h	4 h

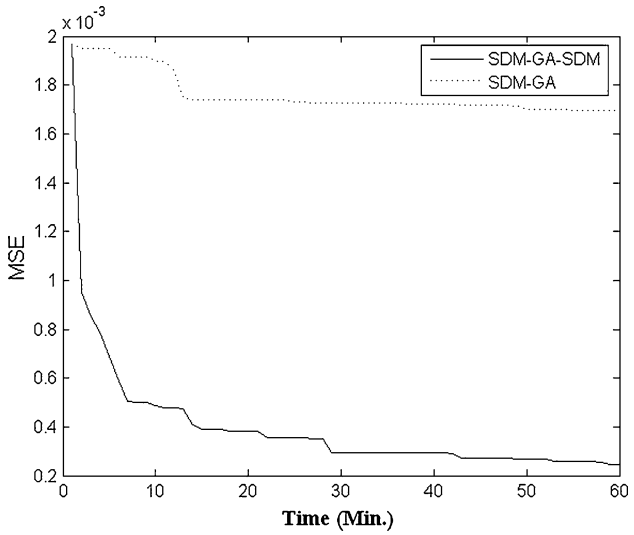

Fig. 5 Comparison of convergence for different combination of GA and SDM

in Fig. 6. The Altera develop board DE2-70 is used for this experiment. The CPU is 100MHZ, and a SOC Builder is used for connecting the audio codec required for

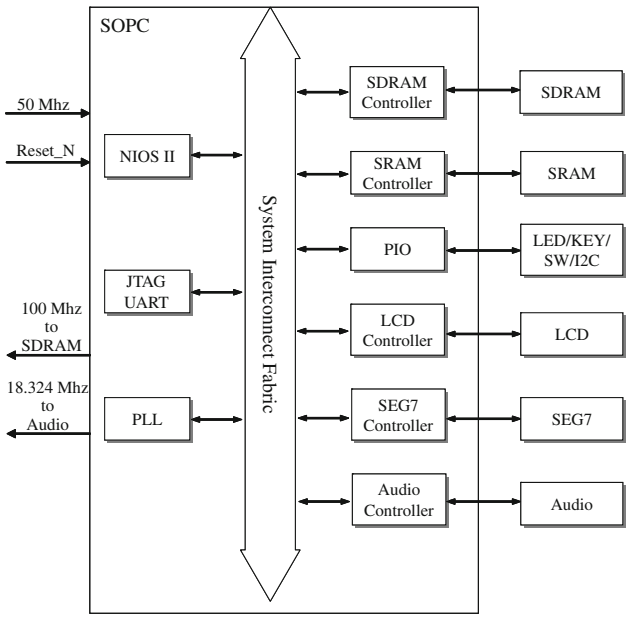

Fig. 6 Block diagram of the SOC system in this experiment

Table 5 Speech recognition rate in PC and FPGA-based SOPC

Speech	Recognition rate		Speech	Recognition rate		Average	
	PC	FPGA		PC	FPGA	PC	FPGA
0	0.95	0.84	5	1	0.95	0.95	0.919
1	1	0.95	6	0.9	0.94		
2	1	0.97	7	0.9	0.75		
3	0.7	0.96	8	1	0.98		
4	1	0.96	9	1	0.89		

measurement of speech signal. The software Nios II is used for compiling the program. As for the hardware, SRAM and flash RAM are used for storing the source code and testing speech, respectively. I^2C Protocol is used to control the register of the platform. Besides, audio controller is used to measure the speech, and SEG7 is used for displaying the recognition results. The standard control IPs, which are supported by SOC Builder, are used to drive the elements SDRAM, SRAM and LCD. In the experiment, PLL has a frequency of 100 MHz and a delay of 3 ns.

The recognition rate in the PC and FPGA-based SOPC is shown in Table 5. It is noted that since the fixed-point arithmetic is used in the FPGA-based SOPC hardware, the computation ability of this hardware is weaker than that of

Table 6 Computation time for speech recognition by HMM [5]/ANN on FPGA

Process	HMM (Avg.)	ANN (Avg.)
Pre-processes	0.109 s.	
MFCC	0.464 s.	
Recognition	0.747 s.	0.04770 s.

PC. This makes the recognition rate in this hardware lower than that of PC. However, in this paper, the downgrade in recognition rate is only about 3 %. This implies that the hardware in this paper is well implemented. Besides, the time for speech recognition by HMM [5] and ANN on FPGA-based SOPC is compared in Table 6. Table 6 reveals that the time for speech recognition by ANN is much less than that by HMM. Hence, ANN is more suitable than HMM to be implemented on a chip.

6 Conclusions

Since the characteristics of SDM and GA complement each other, this paper makes good use of these two simple and popular algorithms and then proposes an HLA architecture to train ANN for speech recognition. The experiment results show that the proposed HLA outperforms other popular algorithms. Moreover, the trained ANN-based speech recognition system is implemented on an FPGA-based chip. According to the hardware experimental results, it can be concluded that ANN is more suitable than SVM or HMM to be implemented on a chip.

Acknowledgments This research work was supported by the National Science Council of the Republic of China under contract NSC 100-2221-E-390-025-MY2.

References

1. Sivaram GSVS, Nemala SK, Mesgarani N, Hermansky H (2010) Data-driven and feedback based spectro-temporal features for speech recognition. IEEE Signal Process Lett 17(11):957–960
2. Lauria S (2007) Talking to machines: introducing Robot perception to resolve speech recognition uncertainties. Circuits Syst Signal Process 26(4):513–526
3. Wan CY, Lee LS (2008) Histogram-based quantization for robust and/or distributed speech recognition. IEEE Trans Audio Speech Lang Processing 16(4):859–873
4. Hagon MT, Demuth HB, Beale M (1996) Neural network design. Thomson Learning, Stamford
5. Kwong S, Chau CW (1997) Analysis of parallel genetic algorithms on HMM based speech recognition system. IEEE Trans Consumer Electron 43(4):1229–1233
6. Shi Y, Liu J, Liu R (2001) Single-chip speech recognition system based on 8051 microcontroller core. IEEE Trans Consumer Electron 47(1):149–153
7. Lin FJ, Huang PK, Chou WD (2007) Recurrent-fuzzy-neural-network-controlled linear induction motor servo drive using genetic algorithms. IEEE Trans Ind Electron 54(3):1449–1461
8. Karamalis PD, Kanatas AG, Constantinou P (2009) A genetic algorithm applied for optimization of antenna arrays used in mobile radio channel characterization devices. IEEE Trans Instrum Meas 58:2475–2487
9. Chu WC (2003) Speech coding algorithms. Wiley, Wiley-IEEE, New Jersey
10. Huang X, Acero A, Wuenon H (2005) Spoken language processing a guide to theory algorithm and system development. Pearson, London
11. Leung HF, Lam HK, Ling SH (2003) Tuning of the structure and parameters of a neural network using an improved genetic algorithm. IEEE Trans Neural Netw 14:79–88
12. Kennedy J, Eberhart RC (1995) "Particle swarm optimization." In: Proceedings IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, pp IV:1942–1948, 1995
13. Storn R, Price K (1997) Differential evolution- A simple and efficient heurist for global optimization over continuous spaces. J. of Global Optimization 11:341–359
14. Runstein F, Violaro F (1995)"An isolated-word speech recognition system using neural networks".In: Proceeding of the 38th midwest symposium on circuit and systems, Vol 1, pp 550–553, 1995
15. Sadaoki F, Dekker M (2001) Digital speech processing, synthesis, and recognition. Marcel Dekker, New York

View publication stats